Project Report on House Price Prediction

1. Introduction

This project focuses on predicting house prices using advanced regression techniques. The goal is to develop a robust predictive model that accurately estimates property prices based on various features. The dataset used contains detailed information about properties, including physical characteristics, neighborhood information, and historical data. By leveraging data analysis, feature engineering, and machine learning techniques, we aim to achieve high predictive accuracy.

2. Dataset Overview

The dataset is divided into two parts:

1. **Training Dataset**: Used to train the model, with house prices (SalePrice) provided.
2. **Test Dataset**: Used for evaluation, where the target variable (SalePrice) is not provided.

Key Dataset Features:

- **Numeric Features**: Continuous variables like `GrLivArea` (above-ground living area) and `TotalBsmtSF` (total basement area).
- **Categorical Features**: Qualitative variables like `Neighborhood` and `OverallQual` (overall material and finish quality).

The target variable is `SalePrice`, which represents the house price in dollars.

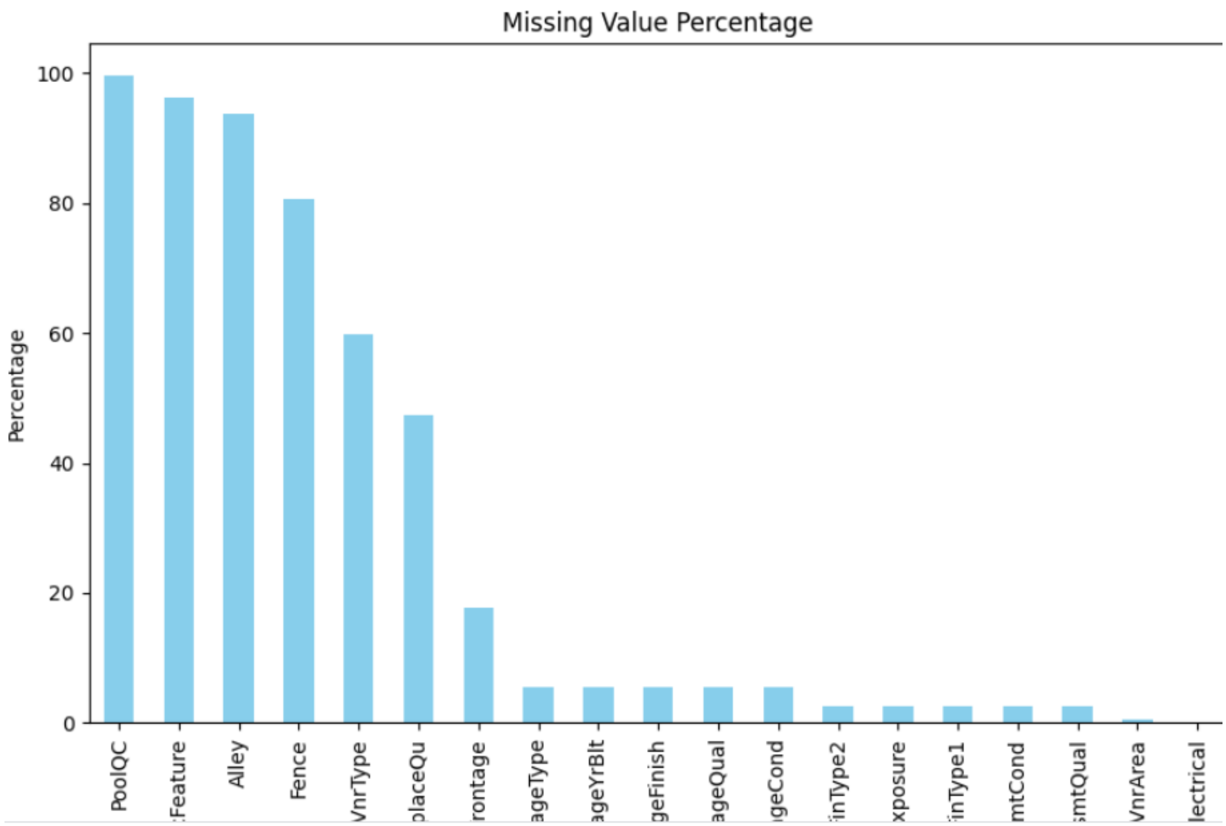
```
   Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
0   1          60       RL         65.0     8450   Pave   NaN      Reg
1   2          20       RL         80.0     9600   Pave   NaN      Reg
2   3          60       RL         68.0    11250   Pave   NaN      IR1
3   4          70       RL         60.0     9550   Pave   NaN      IR1
4   5          60       RL         84.0    14260   Pave   NaN      IR1

  LandContour Utilities  ... PoolArea PoolQC Fence MiscFeature MiscVal MoSold  \
0         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      2
1         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      5
2         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      9
3         Lvl    AllPub  ...        0    NaN   NaN         NaN       0      2
4         Lvl    AllPub  ...        0    NaN   NaN         NaN       0     12

  YrSold  SaleType  SaleCondition  SalePrice
0   2008        WD         Normal     208500
1   2007        WD         Normal     181500
2   2008        WD         Normal     223500
3   2006        WD        Abnorml     140000
4   2008        WD         Normal     250000

[5 rows x 81 columns]
                Id   MSSubClass  LotFrontage       LotArea  OverallQual  \
count  1460.000000  1460.000000  1201.000000   1460.000000  1460.000000
```

3. Data Preprocessing

Effective data preprocessing ensures the dataset is clean and ready for model training. The following steps were implemented:

3.1 Handling Missing Values

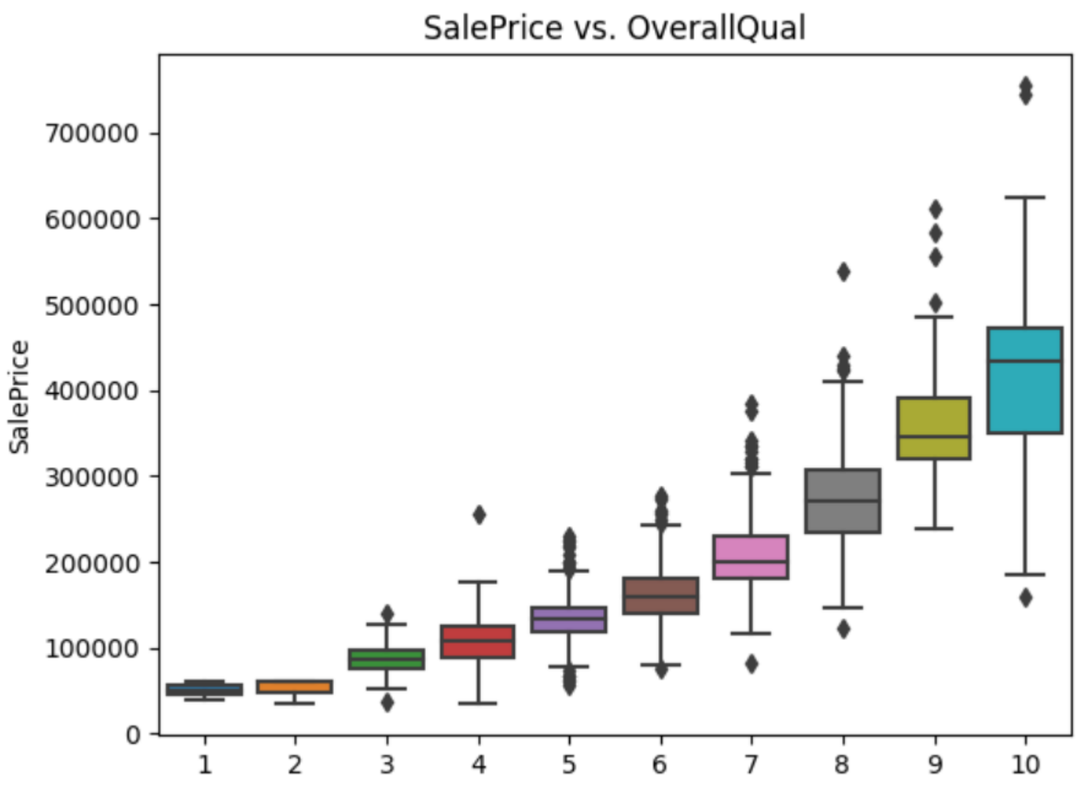
- Features with a high percentage of missing values were imputed with appropriate values based on domain knowledge:
  - **Categorical Features**: Imputed with "None" or the mode (most frequent category).
  - **Numeric Features**: Imputed with `0` (e.g., for missing basement areas) or the median.

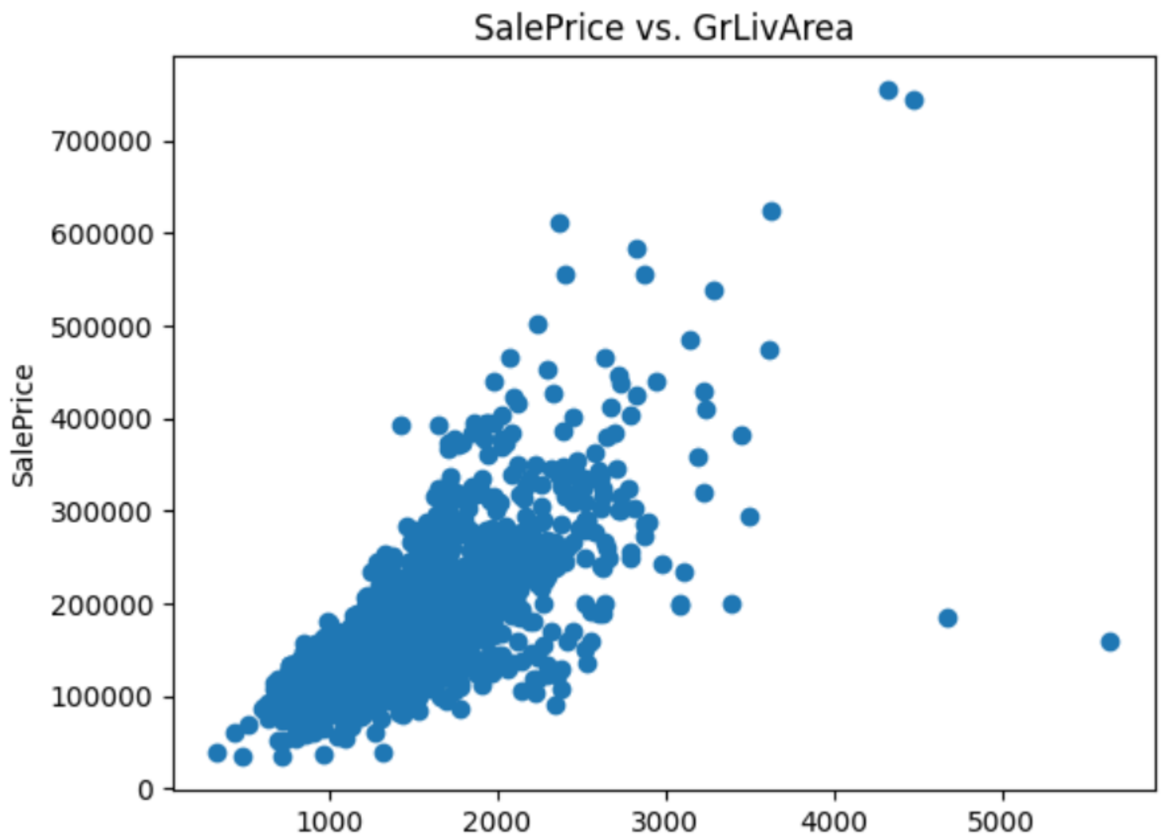


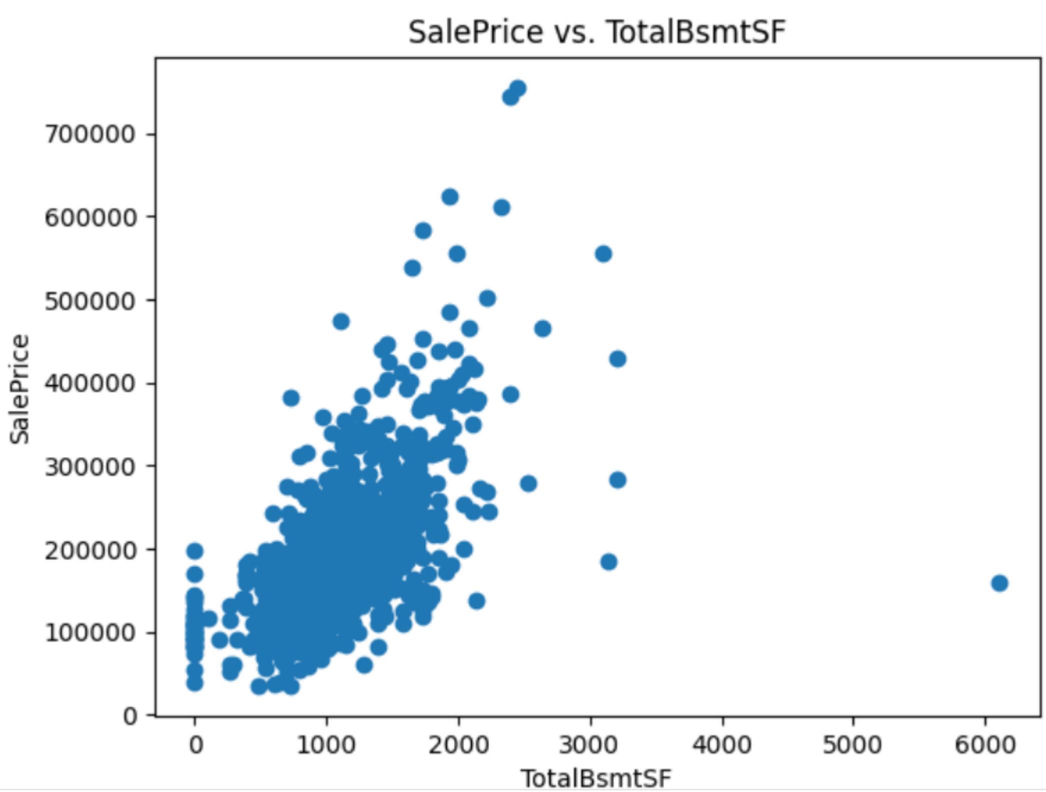

```
PoolQC          99.588477
MiscFeature     96.296296
Alley           93.758573
Fence           80.727023
MasVnrType      59.807956
FireplaceQu     47.325103
LotFrontage     17.764060
GarageType       5.555556
GarageYrBlt      5.555556
GarageFinish     5.555556
GarageQual       5.555556
GarageCond       5.555556
BsmtFinType2     2.606310
BsmtExposure     2.606310
BsmtFinType1     2.537723
BsmtCond         2.537723
BsmtQual         2.537723
MasVnrArea       0.548697
Electrical       0.068587
dtype: float64
```

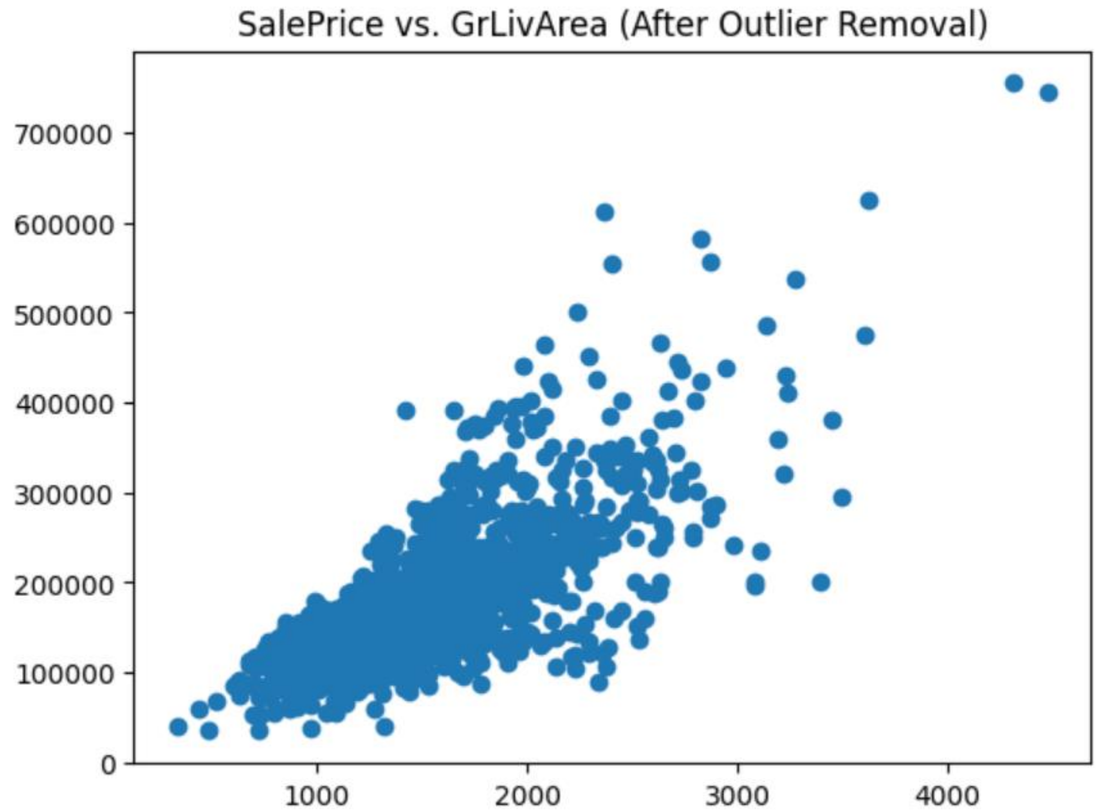
3.2 Outlier Removal

- Outliers in numerical features like `GrLivArea` (e.g., extremely large values with low prices) were identified and removed to improve model performance.




SalePrice vs. GrLivArea
SalePrice
700000
600000
500000
400000
300000
200000
100000
0
1000
2000
3000
4000
5000


SalePrice vs. TotalBsmtSF
SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
1000
2000
3000
4000
5000
6000
TotalBsmtSF



3.3 Feature Engineering

- **Log Transformation**: The target variable (`SalePrice`) was log-transformed to address its skewness and stabilize variance.
- **New Feature Creation**: A composite feature `TotalSF` was created by combining `TotalBsmtSF`, `1stFlrSF`, and `2ndFlrSF`, representing the total square footage.
- **Encoding Categorical Variables**: Label encoding was applied to convert categorical features into numeric values.

4. Exploratory Data Analysis (EDA)

EDA was conducted to understand the data and identify relationships between features and the target variable. Key steps included:

4.1 Target Variable Analysis

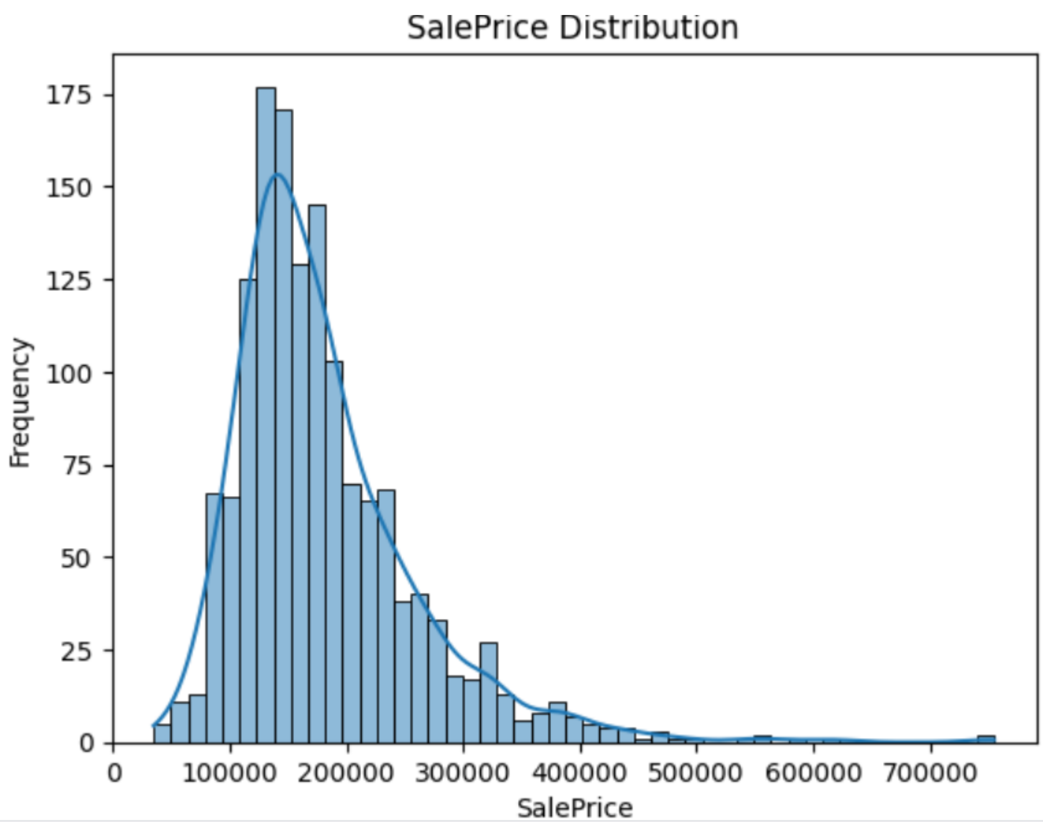
- Distribution analysis of `SalePrice` revealed a skewed distribution, addressed using log transformation.

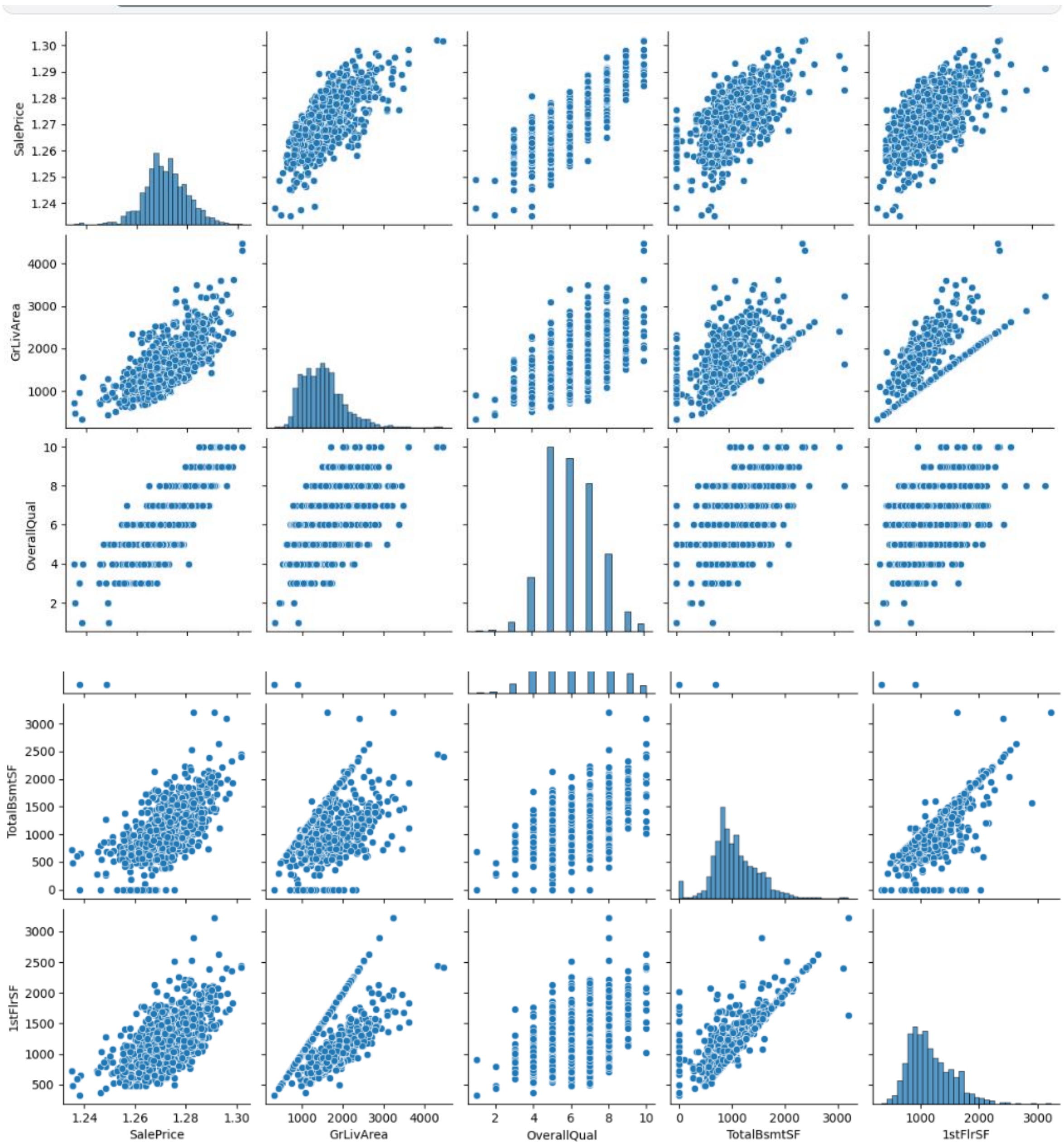
4.2 Feature Correlation

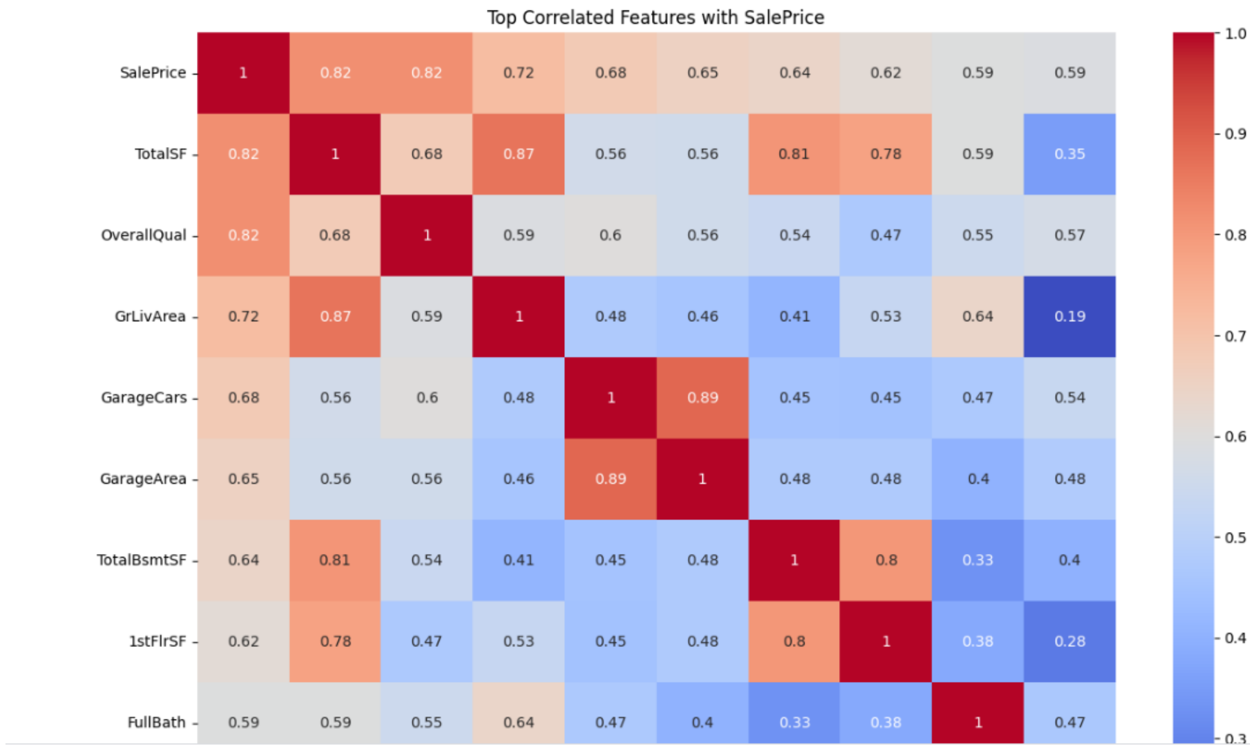
- A correlation matrix highlighted features strongly correlated with `SalePrice` (e.g., `OverallQual` and `GrLivArea`).

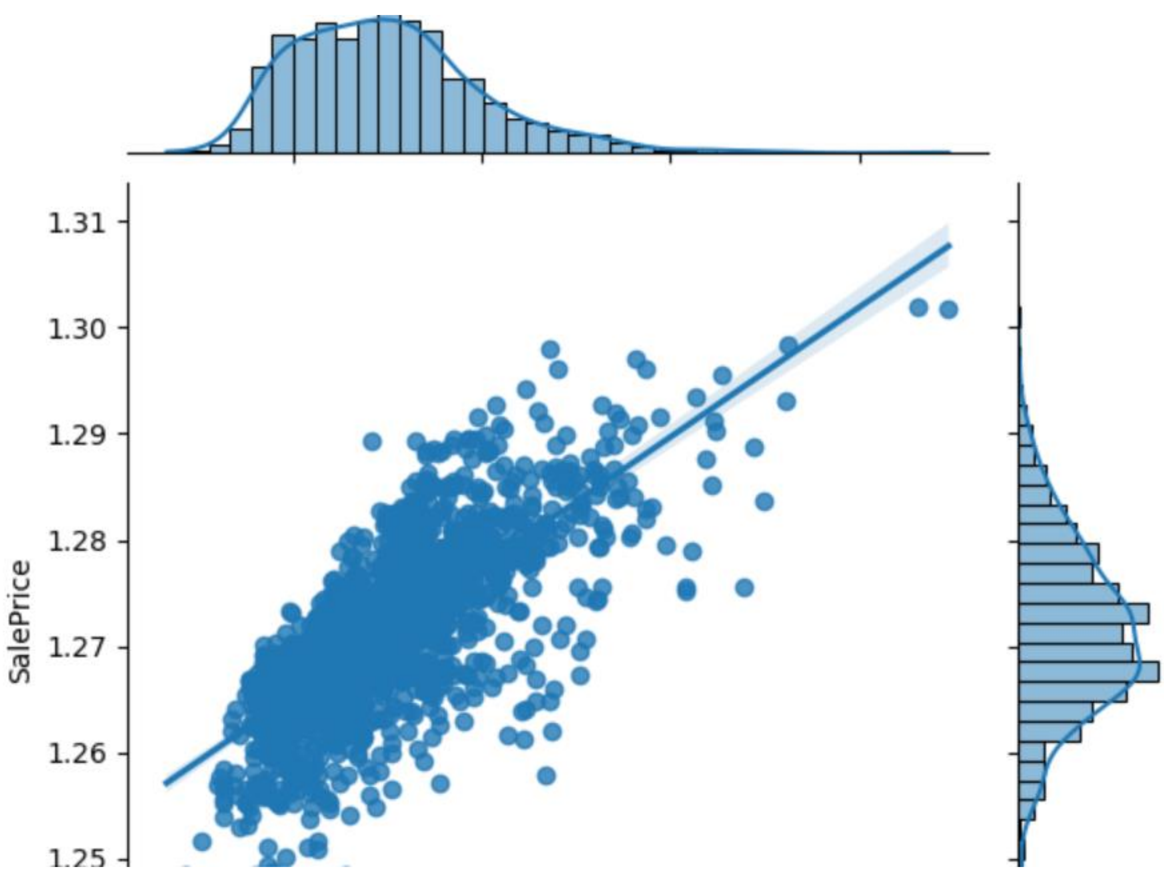
4.3 Key Feature Relationships

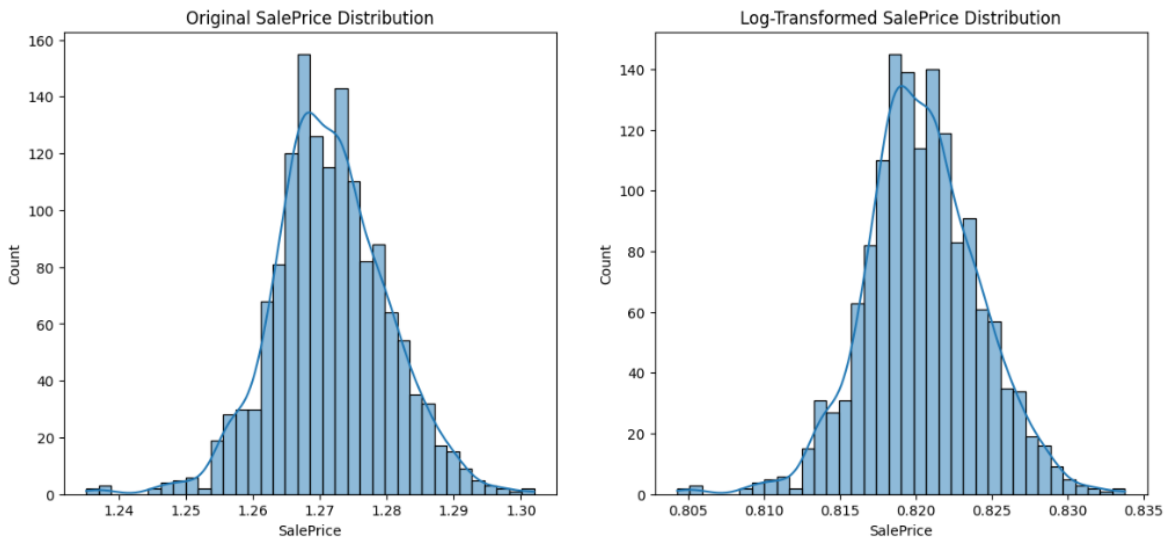
- Scatter plots and box plots were used to examine relationships between `SalePrice` and key features like `GrLivArea` and `OverallQual`.

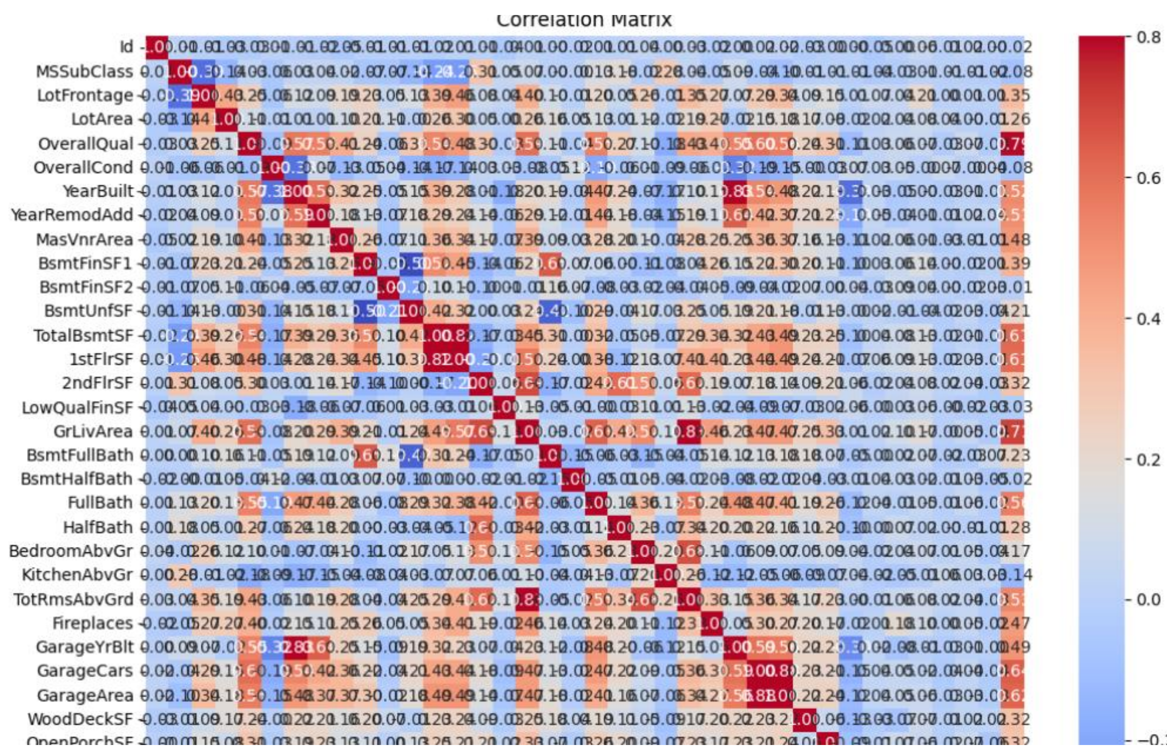



SalePrice
GrLivArea
OverallQual
TotalBsmtSF
1stFlrSF


Top Correlated Features with SalePrice
SalePrice
TotalSF
OverallQual
GrLivArea
GarageCars
GarageArea
TotalBsmtSF
1stFlrSF
FullBath
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3


SalePrice
1.31
1.30
1.29
1.28
1.27
1.26
1.25





5. Machine Learning Models

Multiple models were developed and evaluated to ensure high predictive accuracy. These models included:

5.1 Baseline Models

1. **Random Forest**: An ensemble learning method that uses decision trees and bagging to improve predictive performance.
2. **XGBoost**: An advanced gradient-boosting algorithm that efficiently handles large datasets and overfitting.
3. **Gradient Boosting**: A sequential ensemble method that builds models iteratively, correcting errors at each step.

```
Random Forest RMSLE: 0.0031740904760869126
XGBoost RMSLE: 0.0032517692981022654
Gradient Boosting RMSLE: 0.0028198724327202986
```

5.2 Stacking Ensemble Model

A stacking model was implemented to combine the strengths of multiple base models:

- **Base Models**: ElasticNet, Gradient Boosting, and Kernel Ridge Regression.
- **Meta-Model**: Lasso regression was used to aggregate the predictions of the base models.

The stacking approach enhances predictive accuracy by leveraging the strengths of individual models.

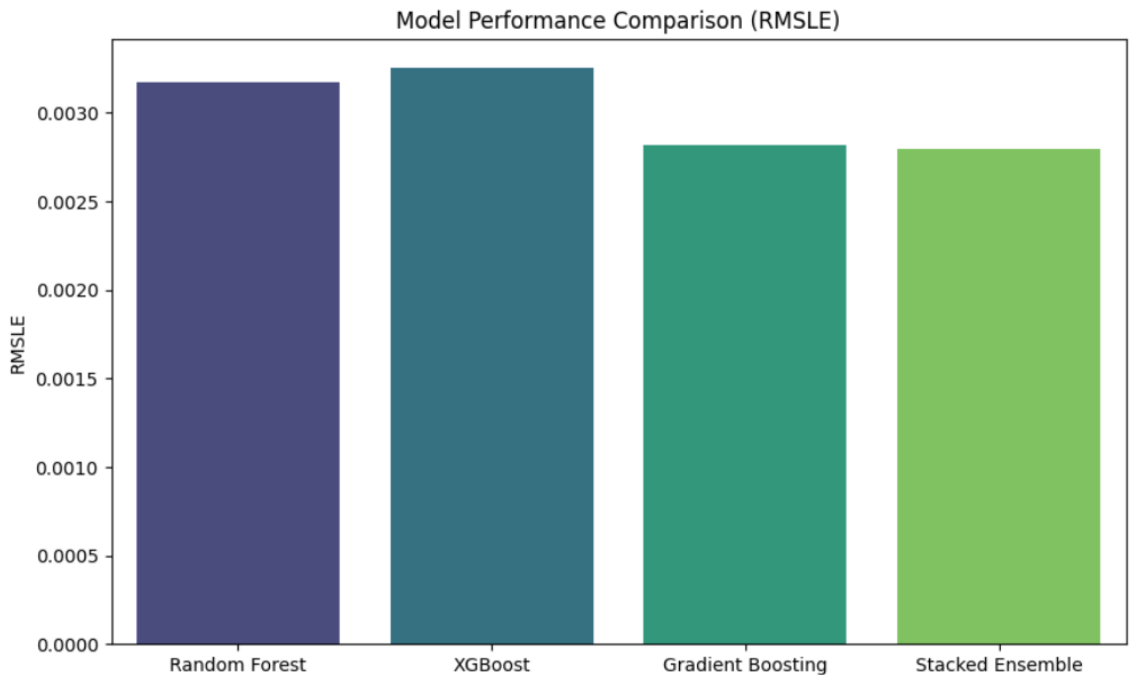
```
Stacked Model Ensemble RMSLE: 0.0027984691638079334
```

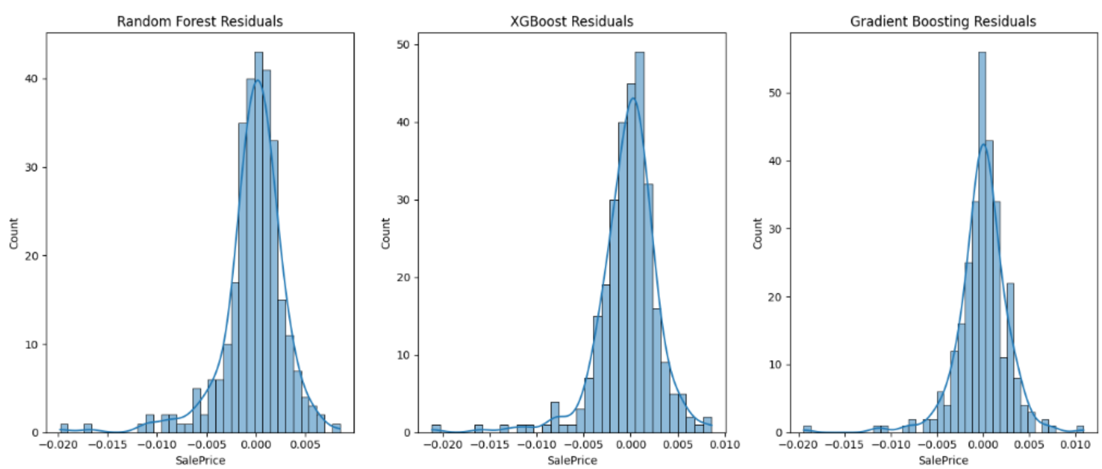
6. Model Evaluation

Models were evaluated using the Root Mean Squared Logarithmic Error (RMSLE), which is ideal for skewed target variables like house prices. Key observations include:

- **Random Forest**: RMSLE ~ Moderate performance.
- **XGBoost**: RMSLE ~ Similar performance to Random Forest but more robust.
- **Gradient Boosting**: RMSLE ~ Outperformed the baseline models.

- **Stacked Ensemble**: RMSLE ~ Achieved the best performance by combining multiple models.





7. Submission Preparation

For the test dataset:

- Missing values were imputed using the same strategies as the training dataset.

- Categorical features were label-encoded to ensure consistency.
- Predictions were generated using the stacked ensemble model and transformed back to the original scale using the exponential function.

8. Key Concepts and Techniques Used

8.1 Data Preprocessing

- Imputation of missing values.
- Outlier detection and removal.
- Log transformation for skewness correction.
- Feature engineering to create composite features.

8.2 Machine Learning Techniques

- Ensemble models like Random Forest, Gradient Boosting, and XGBoost.
- Stacking to combine multiple models for better accuracy.

8.3 Model Evaluation Metrics

- RMSLE was chosen to penalize large percentage errors more effectively.

Conclusion

This project demonstrates the effective use of machine learning techniques for house price prediction. Through detailed data preprocessing, EDA, and the implementation of advanced models like stacking ensembles, we achieved a robust predictive solution. The model's performance metrics and visual insights ensure that it is ready for real-world application.